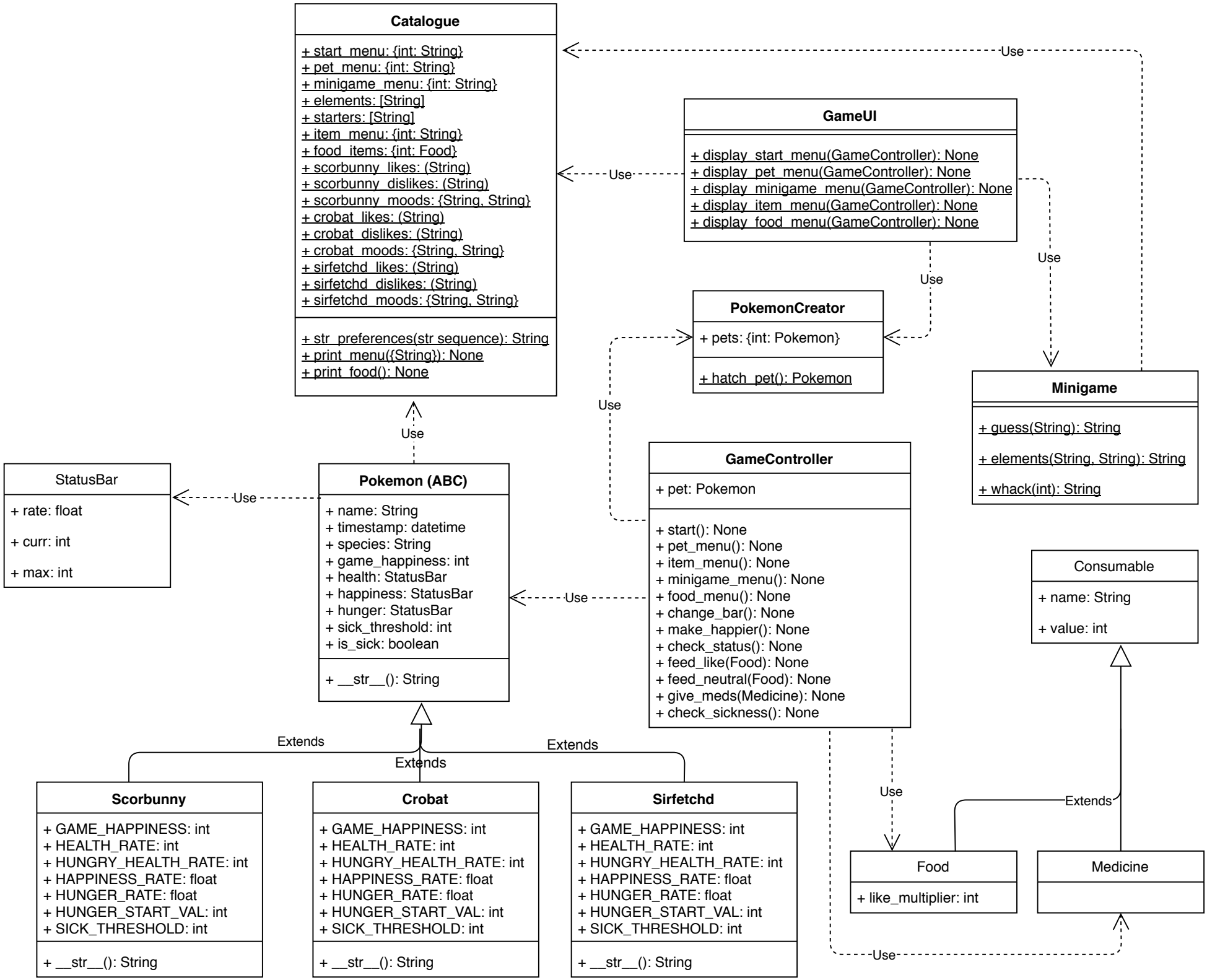
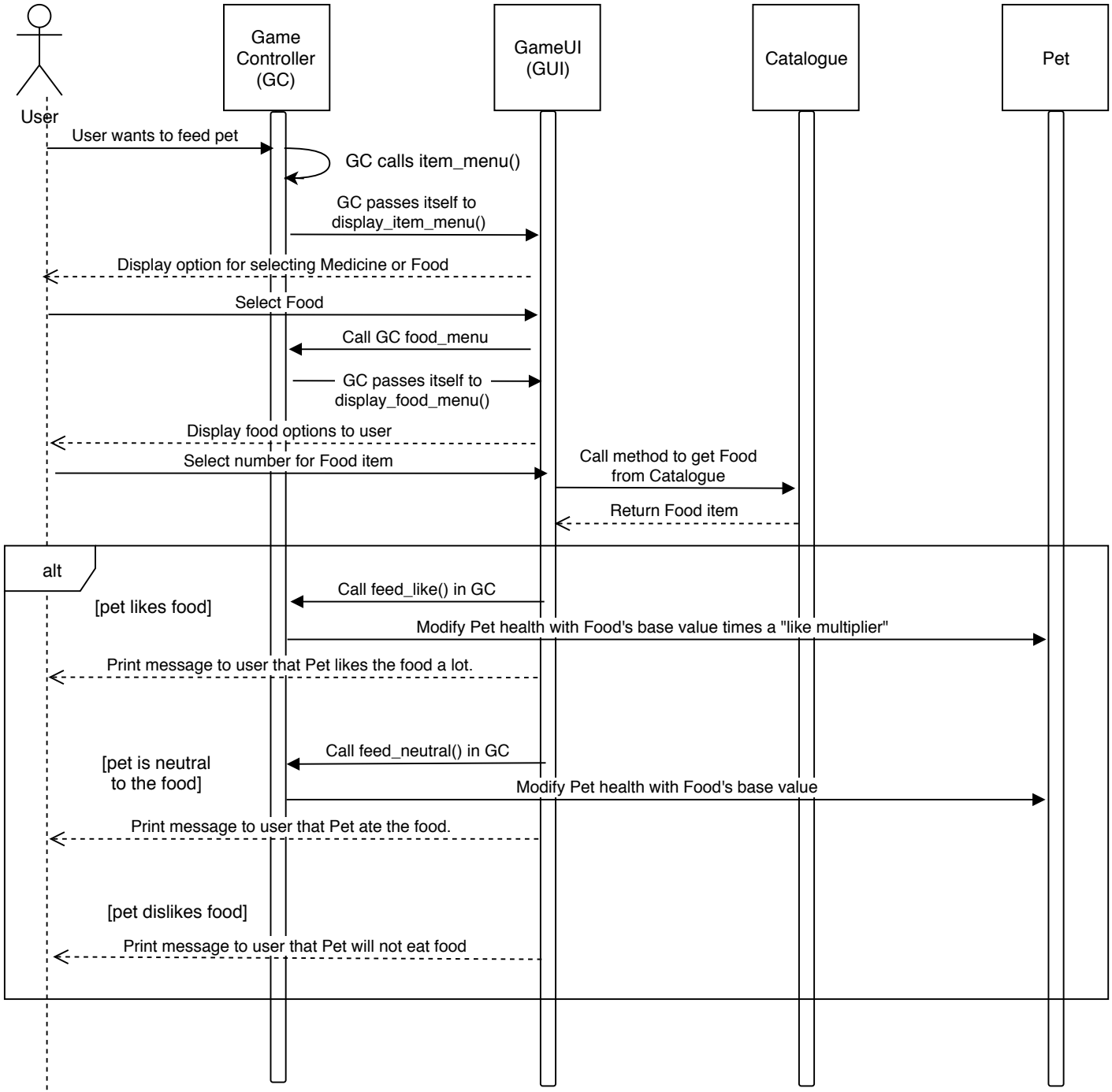

Catalogue
+ start_menu: {int: String}
+ pet_menu: {int: String}
+ minigame_menu: {int: String}
+ elements: [String]
+ starters: [String]
+ item_menu: {int: String}
+ food_items: {int: Food}
+ scorbunny_likes: (String)
+ scorbunny_dislikes: (String)
+ scorbunny_moods: {String, String}
+ crobat_likes: (String)
+ crobat_dislikes: (String)
+ crobat_moods: {String, String}
+ sirfetchd_likes: (String)
+ sirfetchd_dislikes: (String)
+ sirfetchd_moods: {String, String}
+ str_preferences(str sequence): String
+ print_menu({String}): None
+ print_food(): None
Use
GameUI
+ display_start_menu(GameController): None
+ display_pet_menu(GameController): None
+ display_minigame_menu(GameController): None
+ display_item_menu(GameController): None
+ display_food_menu(GameController): None
Use
Use
Use
PokemonCreator
+ pets: {int: Pokemon}
+ hatch_pet(): Pokemon
Minigame
+ guess(String): String
+ elements(String, String): String
+ whack(int): String
Use
Use
StatusBar
+ rate: float
+ curr: int
+ max: int
Use
Pokemon (ABC)
+ name: String
+ timestamp: datetime
+ species: String
+ game_happiness: int
+ health: StatusBar
+ happiness: StatusBar
+ hunger: StatusBar
+ sick_threshold: int
+ is_sick: boolean
+ __str__(): String
Use
GameController
+ pet: Pokemon
+ start(): None
+ pet_menu(): None
+ item_menu(): None
+ minigame_menu(): None
+ food_menu(): None
+ change_bar(): None
+ make_happier(): None
+ check_status(): None
+ feed_like(Food): None
+ feed_neutral(Food): None
+ give_meds(Medicine): None
+ check_sickness(): None
Consumable
+ name: String
+ value: int
Extends
Extends
Extends
Scorbunny
+ GAME_HAPPINESS: int
+ HEALTH_RATE: int
+ HUNGRY_HEALTH_RATE: int
+ HAPPINESS_RATE: float
+ HUNGER_RATE: float
+ HUNGER_START_VAL: int
+ SICK_THRESHOLD: int
+ __str__(): String
Crobat
+ GAME_HAPPINESS: int
+ HEALTH_RATE: int
+ HUNGRY_HEALTH_RATE: int
+ HAPPINESS_RATE: float
+ HUNGER_RATE: float
+ HUNGER_START_VAL: int
+ SICK_THRESHOLD: int
+ __str__(): String
Sirfetchd
+ GAME_HAPPINESS: int
+ HEALTH_RATE: int
+ HUNGRY_HEALTH_RATE: int
+ HAPPINESS_RATE: float
+ HUNGER_RATE: float
+ HUNGER_START_VAL: int
+ SICK_THRESHOLD: int
+ __str__(): String
Use
Extends
Food
+ like_multiplier: int
Medicine
Use


User
Game Controller (GC)
GameUI (GUI)
Catalogue
Pet
User wants to feed pet
GC calls item_menu()
GC passes itself to display_item_menu()
Display option for selecting Medicine or Food
Select Food
Call GC food_menu
GC passes itself to display_food_menu()
Display food options to user
Select number for Food item
Call method to get Food from Catalogue
Return Food item
alt
[pet likes food]
Call feed_like() in GC
Modify Pet health with Food's base value times a "like multiplier"
Print message to user that Pet likes the food a lot.
[pet is neutral to the food]
Call feed_neutral() in GC
Modify Pet health with Food's base value
Print message to user that Pet ate the food.
[pet dislikes food]
Print message to user that Pet will not eat food